

Fondamenti di Informatica e Laboratorio T-AB
Prova Pratica - 7 Luglio 2009
Compito A

Prima di cominciare: si scarichi il file `StartKit5A.zip` contenente i file di esempio.

Avvertenze per la consegna: nominare i file sorgenti come richiesto nel testo del compito, apporre all'inizio di ogni file sorgente un commento contenente i propri dati (**cognome, nome, numero di matricola**) e il **numero** della prova d'esame. Al termine, **consegnare tutti i file sorgente** ed i file contenuti nello StartKit.

Rispettare le specifiche, in particolare inserire le funzioni nei file specificati fra parentesi dopo il nome della funzione. Chi non rispetta le specifiche sarà opportunamente penalizzato. **NON SARANNO CORRETTI** gli elaborati che presenteranno un numero "non ragionevole" di errori di compilazione.

Consiglio: per verificare l'assenza di *warnings*, effettuare di tanto in tanto *Rebuild All*.

Un bagno della riviera romagnola usa un sistema informatico all'avanguardia nella gestione dei clienti. Il sistema permette di tenere traccia delle spese effettuate da ogni singolo cliente, che non paga al momento della consumazione ma solo alla fine del periodo di permanenza. Il sistema registra su un file di testo di nome "`consumi.txt`" le informazioni relative ad ogni consumazione effettuata, in particolare ogni riga rappresenta una consumazione. Su ogni riga, relativamente ad ogni singola consumazione, vengono registrate le seguenti informazioni: il **nome** del cliente (al più 63 caratteri senza spazi), un carattere per indicare il **tipo** di consumazione ('B' per Bar, ad es. caffè e pasta; 'S' per Spiaggia, ad esempio lettino e ombrellone), una stringa di al più 127 caratteri senza spazi indicanti **cosa** è stato consumato (ad esempio, caffè, oppure 1 lettino), e l'**importo** della consumazione (un float).

Quando giunge il momento di pagare, il gestore del bagno calcola manualmente l'importo che un cliente deve pagare: tale operazione risulta lunga, e soprattutto soggetta ai numerosi errori di conto del gestore. Si chiede al candidato di realizzare un programma che calcoli opportunamente, per ogni cliente presente nel file dei consumi, l'importo totale da pagare, avendo cura di distinguere l'importo relativo alle consumazioni da bar (segnate con il carattere 'B') e quelle relative alla spiaggia (distinte tramite il carattere 'S').

Esercizio 1 - Lettura delle consumazioni (`consumi.h/consumi.c`)

Si definisca una struttura dati **consumazione**, al fine di rappresentare i dati memorizzati su una singola riga del file `consumi.txt` (nome, tipo, cosa, importo). Quindi si realizzi una funzione:

```
consumazione leggiUnaConsumazione(FILE * fp);
```

che, ricevuto in ingresso un puntatore ad un file già opportunamente aperto, legga i dati relativi ad una consumazione e li restituisca come struttura dati **consumazione**.

Il candidato realizzi poi una funzione:

```
consumazione * leggiTutteLeConsumazioni(char * nomeFile, int * dim);
```

che, ricevuto come parametro di ingresso il nome di un file contenente le consumazioni, provveda ad aprirlo e ne legga il contenuto. A tal fine, si usi la funzione `leggiUnaConsumazione(...)` appena definita. Si noti che non è noto a priori quante consumazioni siano presenti nel file: sarà quindi necessario determinare quante ve ne siano, e poi allocare memoria dinamicamente nella dimensione opportuna.

Nello `StartKit4A.zip` il candidato troverà a disposizione un file di testo, "`consumi.txt`", contenente una lista di consumazioni di esempio.

Contestualmente, il candidato scriva nel main opportune istruzioni per invocare le funzioni qui definite, al fine di verificarne la corretta implementazione: il candidato abbia cura, dopo aver letto i dati relativi agli studenti, di stampare a video i dati letti al fine di verificarne l'effettiva e corretta corrispondenza con quanto memorizzato in "`consumi.txt`". Una volta verificato il corretto funzionamento delle funzioni, il candidato non cancelli il codice nel main ma si limiti a commentarlo.

Fondamenti di Informatica e Laboratorio T-AB
Prova Pratica - 7 Luglio 2009
Compito A

Esercizio 2 - Definizione della struttura dati "conto" e ricerca (consumi.h/consumi.c)

Al fine di calcolare i totali di ogni cliente, il candidato definisca un'opportuna struttura dati di nome **conto**, che permetta la memorizzazione del **nome** del cliente, **dell'importo** totale delle consumazioni relative al Bar (tipo 'B'), e **dell'importo** totale delle consumazioni relative alla Spiaggia (tipo 'S').

Il candidato realizzi poi una funzione:

```
int cerca(conto * c, int dim, char * nome);
```

che, ricevuto in ingresso un array **c** di strutture dati di tipo **conto** e di dimensione **dim**, cerchi in tale array se esiste già un conto relativo al cliente il cui nome è specificato tramite il parametro **nome**. Qualora esista già una struttura dati relativa al cliente specificato, la funzione deve restituire l'indice nell'array (un valore compreso quindi tra 0 e **dim-1**). Qualora invece non esista ancora una struttura dati relativa a quel cliente, la funzione restituisca il valore -1.

Contestualmente, il candidato scriva nel main opportune istruzioni per invocare le funzioni definite al fine di verificarne la corretta implementazione. Una volta verificato il corretto funzionamento delle funzioni, il candidato non cancelli il codice nel main ma si limiti a commentarlo.

Esercizio 3 - Calcolo dei totali (consumi.h/consumi.c)

Il candidato realizzi la funzione:

```
int calcolaTotali(consumazione * c, int dim_c, conto * t, int dim_t);
```

che riceve in ingresso: un array **c** di strutture dati di tipo **consumazione**, e la sua dimensione **dim_c**; un array **t** di strutture dati di tipo **conto**, e la sua dimensione fisica massima **dim_t**. La funzione deve provvedere a calcolare i totali relativi ad ogni cliente, distinti per tipologia di spesa (Bar o Spiaggia). In particolare, nell'array **t** si devono memorizzare le opportune strutture dati **conto** man mano che vengono calcolate. La funzione restituisce il numero di strutture dati memorizzate effettivamente in **t**. Si assuma che al massimo i clienti registrati nel file **consumi.txt** siano 200.

Suggerimento. La funzione potrebbe operare nel modo seguente: inizialmente l'array **t** ha dimensione logica pari a 0. Per ogni consumazione presente in **c**, si va a verificare usando la funzione **cerca(...)** (Esercizio 2) se esiste già una struttura dati **conto** memorizzata in **t** e relativa a tale cliente. Se esiste già, si aggiornano i totali aggiungendo la nuova spesa; altrimenti si inizializza una struttura dati **conto** nella prima posizione disponibile nell'array **t**, e si incrementa la dimensione logica dell'array.

Esercizio 4 - Ordinamento dei clienti e dei totali (consumi.h/consumi.c)

Si realizzi una procedura che, ricevuto in ingresso un array di strutture dati **conto**, ordini tale array in base all'importo totale (relativo alle consumazioni da bar ed alle consumazioni da spiaggia), che il cliente deve pagare, in ordine decrescente.

```
void ordina(conto * t, int dim);
```

A tal scopo, si utilizzi uno qualunque degli algoritmi di ordinamento visti a lezione, avendo cura di modificarlo opportunamente per il caso in questione.

Esercizio 5 - Main (main.c)

Il candidato realizzi un programma **main.c** che provveda a leggere le consumazioni dal file "**consumi.txt**" e a memorizzarli tramite la funzione di cui al punto 1. Quindi il programma allochi spazio sufficiente per memorizzare un array di strutture dati **conto** (inizialmente vuoto), e tramite la funzione di cui al punto 3 calcoli quanto deve pagare ogni cliente. Tramite la funzione di cui al punto 4 venga ordinato opportunamente l'array, ed infine stampi a video l'elenco dei clienti con le cifre che questi devono pagare.

Il candidato abbia cura di deallocare (al termine del programma) tutte le strutture allocate dinamicamente.

Fondamenti di Informatica e Laboratorio T-AB
Prova Pratica - 7 Luglio 2009
Compito A

"consumi.h":

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
#ifndef CONSUMI
#define CONSUMI
```

```
#define DIM_NOME 64
#define DIM_BENE 128
```

```
typedef struct {
    char nome[DIM_NOME];
    char cosa[DIM_BENE];
    char tipo;
    float importo;
}
consumazione;
```

```
typedef struct {
    char nome[DIM_NOME];
    float importo_B;
    float importo_S;
}
conto;
```

```
#endif
```

```
consumazione leggiUnaConsumazione(FILE * fp);
void stampaUnaConsumazione(consumazione c);
consumazione * leggiTutteLeConsumazioni(char * nomeFile, int * dim);

int cerca(conto * c, int dim, char * nome);

int calcolaTotali(consumazione * c, int dim_c, conto * t, int dim_t);

void ordina(conto * t, int dim);
```

Fondamenti di Informatica e Laboratorio T-AB
Prova Pratica - 7 Luglio 2009
Compito A

```
"consumi.c":
#include "consumi.h"

consumazione leggiUnaConsumazione(FILE * fp) {
    consumazione result;
    fscanf(fp, "%s %c %s %f", result.nome, &(result.tipo), result.cosa,
&(result.importo) );
    return result;
}

void stampaUnaConsumazione(consumazione c) {
    printf("Nome: %s\n", c.nome);
    printf("Tipo: %c\n", c.tipo);
    printf("Cosa: %s\n", c.cosa);
    printf("Importo: %6.2f\n", c.importo);
}

consumazione * leggiTutteLeConsumazioni(char * nomeFile, int * dim) {
    consumazione * result = NULL;
    consumazione temp;
    FILE * fp;
    int i;

    *dim = 0;
    if ((fp=fopen(nomeFile, "r")) == NULL) {
        *dim = -1;
        return result;
    }
    while (fscanf(fp, "%s %c %s %f", temp.nome, &(temp.tipo), temp.cosa,
&(temp.importo)) == 4)
        *dim = *dim + 1;
    rewind(fp);

    result = (consumazione *) malloc(sizeof(consumazione) * *dim);
    for (i=0; i<*dim; i++)
        result[i] = leggiUnaConsumazione(fp);
    fclose(fp);
    return result;
}

int cerca(conto * c, int dim, char * nome) {
    int i;
    int trovato;

    trovato = 0;
    i=0;
    while (i<dim && !trovato) {
        if (strcmp(c[i].nome, nome)==0)
            trovato = 1;
        if (!trovato)
            i = i+1;
    }
    if (trovato)
        return i;
    else
        return -1;
}

int calcolaTotali(consumazione * c, int dim_c, conto * t, int dim_t) {
    int pos;
    int i;
    int dim = 0;
```

Fondamenti di Informatica e Laboratorio T-AB
Prova Pratica - 7 Luglio 2009
Compito A

```
for (i=0; i<dim_c; i++) {
    pos = cerca(t, dim, c[i].nome);
    if (pos>=0) {
        if (c[i].tipo == 'B')
            t[pos].importo_B += c[i].importo;
        else
            if (c[i].tipo == 'S')
                t[pos].importo_S += c[i].importo;
            else {
                printf("Errore: consumazione non riconosciuta!\n");
                stampaUnaConsumazione(c[i]);
            }
    }
    else {
        strcpy(t[dim].nome, c[i].nome);
        if (c[i].tipo == 'B') {
            t[dim].importo_B = c[i].importo;
            t[dim].importo_S = 0;
        }
        else
            if (c[i].tipo == 'S') {
                t[dim].importo_S = c[i].importo;
                t[dim].importo_B = 0;
            }
            else {
                printf("Errore: consumazione non riconosciuta!\n");
                stampaUnaConsumazione(c[i]);
            }
        dim++;
    }
}
return dim;
}

void scambia(conto * a, conto * b) {
    conto temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

void bubbleSort(conto v[], int n) {
    int i;
    int ordinato = 0;

    while (n>1 && ordinato==0) {
        ordinato = 1;
        for (i=0; i<n-1; i++) {
            if ( (v[i].importo_B+v[i].importo_S) <
                (v[i+1].importo_B+v[i+1].importo_S) ) {
                scambia( &v[i], &v[i+1]);
                ordinato = 0;
            }
        }
        n--;
    }
}

void ordina(conto * t, int dim) {
    bubbleSort(t, dim);
}
```

Fondamenti di Informatica e Laboratorio T-AB
Prova Pratica - 7 Luglio 2009
Compito A

"main.c":

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#include "consumi.h"

int main(void)
{
    int i;
    int dim = 0;
    consumazione * elenco;
    conto totali[200];
    int dim_t = 0;

    elenco = leggiTutteLeConsumazioni("consumi.txt", &dim);
    for (i=0; i<dim;i++)
        stampaUnaConsumazione(elenco[i]);

    {
        conto temp[1];
        conto c1 ={"chesani", 0, 0};
        temp[0] = c1;
        printf("Posizione: %d\n", cerca(temp, 1, "chesani"));
        printf("Posizione: %d\n", cerca(temp, 1, "milano"));
    }

    dim_t = calcolaTotali(elenco, dim, totali, 200);
    ordina(totali, dim_t);
    for (i=0; i<dim_t; i++) {
        printf("cliente: %s\n", totali[i].nome);
        printf("bar: %6.2f\n", totali[i].importo_B);
        printf("spiaggia: %6.2f\n", totali[i].importo_S);
    }

    free(elenco);
    system("PAUSE");

    return (0);
}
```